GIT and GitHub Tutorials (Anuj Bhaiya):

* On installing Git first time, open Git Bash and type below commands:

CTS+2262969@LTIN391270 MINGW64 ~

$ git --version

git version 2.37.3.windows.1

CTS+2262969@LTIN391270 MINGW64 ~

$ git config --global user.name "Nilay Adhikary"

CTS+2262969@LTIN391270 MINGW64 ~

$ git config --global user.email "Nilay.Adhikary@cognizant.com"

CTS+2262969@LTIN391270 MINGW64 ~

$ git config user.email

Nilay.Adhikary@cognizant.com

If all is correct, git will show your name and email address.

* git status

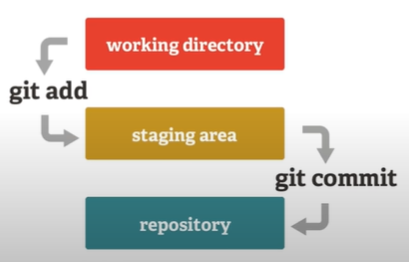
This will give you the status of your files, if there is any changes happened in the current repository, check for untracked files, etc. Untracked files are yet to be committed.

In git, we first track a file, and then commit that file.

* **How to stage a file (before committing on that file):**

git add <filename> for e.g. git add sum.js

This will stage sum.js file. Staging area is a place where you hold the changes before committing those. For example, if your project has multiple files and you want to commit only on some of the files, then you move those files in staging area.



* **How to commit for the first time:**

**git commit -m "Initial Commit"**

For every time you commit, you can set your message as above. H

* **How to show the details of previous commits?**

**git log**

This will show all the previous commits details with the messages and by whom the commits were done.

* **How to send all the modified and untracked files from a folder to the staging area and commit them together?**

**git add .**

this will send all the files in staging area and

**git commit -m “All files have been commited”**

this will commit all the changes.

------------------------------- Checkout command ----------------------------------------

Basically, with checkout command, we go from one branch to another branch.

* **Now suppose you want to go back to your initial commit. How to go back to some other commit?**

**git checkout <commit hash code/branch name>**

All other files make after this commit will be deleted or unavailable after this.

* **How to get back to the master branch from this?**

To get back to the current position(branch), use like this:

**git checkout <branch name where you want to go back>**

If we want to go to the master branch put that. And all the code files that were unavailable, will be available again.

**What is branch?**

In production level projects, there may be multiple people doing work on same project but assigned with different tasks. So, to maintain that we use different branch for different developers.

Then we merge all the sub-branches into the one master branch, which is ultimately the project ready to deploy.

Tasks in different branches work independently. You can basically build multiple branch(es) from one branch, and it forms a tree-like structure.

A screenshot of a computer

Description automatically generated

* **Now let’s see how we can make different branch from master branch.**

**git branch <branch name>**

**e.g. git branch dev**

Now a new branch dev has already added. To check that out write

**git branch**

It will show two branch names master and dev.

* **Now to switch to dev branch type**

**git checkout dev**

Now we have switched to dev branch and adding files to this branch will not affect the other master branch.

* **Now to create a new branch and switch to it directly from dev branch requires this:**

**git checkout -b <branch name>**

**e.g. git checkout -b “Nilay/multiply”**

Now we have created a new branch Nilay/multiply and switched to that one.

Now the branch structure is master 🡪 dev 🡪 Nilay/multiply.

* **Now to merge the Nilay/multiply branch (after the completion of the tasks):**

First come to the branch where you want to merge another branch. For e.g.

to merge Nilay/multiply to dev, first come to dev and merge like:

**git checkout dev (comes to dev branch)**

**git merge Nilay/multiply (then merges)**

Now the files those were available only in Nilay/multiply, will also be available in the dev branch.